EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	21018178	@ad<"20010709"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 19:55
L2	1	(Ken near2 Fernald).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 19:56
L3	0	711/163>"ccls."	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 19:56
L4	1351	711/163.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 19:57
L5	878	(lower or upper or higher or portion$2) with (eras$4 with befor$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 19:58
L6	43964	(lock$4 near3 (bit$3 or field$2 or byte$4 or tag$4 or head$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:06
L7	40218	eras$4 with (befor$6 or prior$4 or after$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:08
L8	167234	(request$4 or access$4 or clock$4 or protect$4) near5 (block$2 or row$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:06
L9	38	(lock$4 near3 (bit$3 or field$2 or byte$4 or tag$4 or head$4)) near5 locat$4 near5 read$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:08

EAST Search History

L10	0	5 and 9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:08
L11	0	((lock$4 near3 (bit$3 or field$2 or byte$4 or tag$4 or head$4)) near5 locat$4 near5 read$4) and 5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:09
L12	17055	(logical or virtual) adj2 memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:09
L13	40955	shar$4 and (lock$4 and bit$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:10
L14	2051	12 and 13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:10
L15	0	14 and 5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:10
L16	400	14 and 6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:10
L17	77	16 and 7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:10
L18	69	17 and 8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:10
L20	1	18 and 1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/08/13 20:11

IEEE Xplore RELEASE 2.1

Sun, 13 Aug 2006, 8:22:05 PM EST

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Search Query Display

[redacted]

Recent Search Queries

#1	(((memory protection) and (lock*))<in>metadata)
#2	(lock* and (bit* or field* or byte* or tag* or head*)<IN>metadata)
#3	(logical or virtual) and memory
#4	((lock* bit) and shar*<IN>metadata)
#5	((bit* or field* or byte* or tag* or head*) and lock*<IN>metadata)
#6	((internal memory) and protect*<IN>metadata)
#7	((((memory protection) and (lock*))<in>metadata)) <AND> ((lock* and (bit* or field* or byte* or tag* or head*)<IN>metadata))
#8	((logical or virtual) and memory) <AND> (((lock* bit) and shar*<IN>metadata))
#9	(((bit* or field* or byte* or tag* or head*) and lock*<IN>metadata)) <AND> (((internal memory) and protect*<IN>metadata))
#10	(((((memory protection) and (lock*))<in>metadata)) <AND> ((lock* and (bit* or field* or byte* or tag* or head*)<IN>metadata))) <AND> (((logical or virtual) and memory) <AND> (((lock* bit) and shar*<IN>metadata)))

[redacted]

Indexed by Inspec

Terms used **memory protection logical virtual shar lock bit tag** Found **86** of **184,245**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 86 Result page: **1** 2 3 4 5 next

Relevance scale

1 Cache Memories
Alan Jay Smith
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3
Publisher: ACM Press
Full text available: pdf(4.61 MB) Additional Information: full citation, references, citings, index terms

2 Hive: fault containment for shared-memory multiprocessors
J. Chapin, M. Rosenblum, S. Devine, T. Lahiri, D. Teodosiu, A. Gupta
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles SOSP '95**, Volume 29 Issue 5
Publisher: ACM Press
Full text available: pdf(1.90 MB) Additional Information: full citation, references, citings, index terms

3 A new page table for 64-bit address spaces
M. Talluri, M. D. Hill, Y. A. Khalidi
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles SOSP '95**, Volume 29 Issue 5
Publisher: ACM Press
Full text available: pdf(1.97 MB) Additional Information: full citation, references, citings, index terms

4 Tempest and typhoon: user-level shared memory
S. K. Reinhardt, J. R. Larus, D. A. Wood
April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture ISCA '94**, Volume 22 Issue 2
Publisher: IEEE Computer Society Press, ACM Press
Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, citings, index terms

Future parallel computers must efficiently execute not only hand-coded applications but also programs written in high-level, parallel programming languages. Today's machines

limit these programs to a single communication paradigm, either message-passing or shared-memory, which results in uneven performance. This paper addresses this problem by defining an interface, *Tempest*, that exposes low-level communication and memory-system mechanisms so programmers and compilers can customize polici ...

5 Tempest and typhoon: user-level shared memory

Steven K. Reinhardt, James R. Larus, David A. Wood

August 1998 **25 years of the international symposia on Computer architecture (selected papers)**

Publisher: ACM Press

Full text available: pdf(1.57 MB) Additional Information: full citation, references, index terms

6 Algorithms for scalable synchronization on shared-memory multiprocessors

John M. Mellor-Crummey, Michael L. Scott

February 1991 **ACM Transactions on Computer Systems (TOCS)**, Volume 9 Issue 1

Publisher: ACM Press

Full text available: pdf(3.07 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Busy-wait techniques are heavily used for mutual exclusion and barrier synchronization in shared-memory parallel programs. Unfortunately, typical implementations of busy-waiting tend to produce large amounts of memory and interconnect contention, introducing performance bottlenecks that become markedly more pronounced as applications scale. We argue that this problem is not fundamental, and that one can in fact construct busy-wait synchronization algorithms that induce no memory or interc ...

7 Minos: Control Data Attack Prevention Orthogonal to Memory Model

Jedidiah R. Crandall, Frederic T. Chong

December 2004 **Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture MICRO 37**

Publisher: IEEE Computer Society

Full text available: pdf(255.53 KB) Additional Information: full citation, abstract, citings

We introduce Minos, a microarchitecture that implements Biba's low-water-mark integrity policy on individual words of data. Minos stops attacks that corrupt control data to hijack program control flow but is orthogonal to the memory model. Control data is any data which is loaded into the program counter on control flow transfer, or any data used to calculate such data. The key is that Minos tracks the integrity of all data, but protects control flow by checking this integrity when a program use ...

8 Cache memory performance in a unix enviroment

Cedell Alexander, William Keshlear, Furrokh Cooper, Faye Briggs

June 1986 **ACM SIGARCH Computer Architecture News**, Volume 14 Issue 3

Publisher: ACM Press

Full text available: pdf(2.10 MB) Additional Information: full citation, citings, index terms

9 A memory management unit and cache controller for the MARS system

Feipei Lai, Chyuan-Yow Wu, Tai-Ming Parng

November 1990 **Proceedings of the 23rd annual workshop and symposium on Microprogramming and microarchitecture**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references

For large caches, the interaction between cache access and address translation affects the machine cycle time and the access time to memory. The physically addressed caches slow down the cache access due to the virtual address translation. The virtually addressed caches is faster, but the synonym problem is difficult to handle. By some software constraints and hardware support, our virtually addressed physically tagged caches can achieve the same speed as traditional virtually addressed cac ...

10 Simple but effective techniques for NUMA memory management

W. Bolosky, R. Fitzgerald, M. Scott

November 1989 **ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles SOSP '89**, Volume 23 Issue 5

Publisher: ACM Press

Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, references, citings, index terms

Multiprocessors with non-uniform memory access times introduce the problem of placing data near the processes that use them, in order to improve performance. We have implemented an automatic page placement strategy in the Mach operating system on the IBM ACE multiprocessor workstation. Our experience indicates that even very simple automatic strategies can produce nearly optimal page placement. It also suggests that the greatest leverage for further performance improvement lies in reducing ...

11 A survey of commercial parallel processors

Edward Gehringer, Janne Abullarade, Michael H. Gulyn

September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4

Publisher: ACM Press

Full text available: pdf(2.96 MB) Additional Information: full citation, abstract, citings, index terms

This paper compares eight commercial parallel processors along several dimensions. The processors include four shared-bus multiprocessors (the Encore Multimax, the Sequent Balance system, the Alliant FX series, and the ELXSI System 6400) and four network multiprocessors (the BBN Butterfly, the NCUBE, the Intel iPSC/2, and the FPS T Series). The paper contrasts the computers from the standpoint of interconnection structures, memory configurations, and interprocessor communication. Also, the share ...

12 Fine-grain access control for distributed shared memory

Ioannis Schoinas, Babak Falsafi, Alvin R. Lebeck, Steven K. Reinhardt, James R. Larus, David A. Wood

November 1994 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the sixth international conference on Architectural support for programming languages and operating systems ASPLOS-VI**, Volume 29 , 28 Issue 11 , 5

Publisher: ACM Press

Full text available: pdf(1.20 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper discusses implementations of fine-grain memory access control, which selectively restricts reads and writes to cache-block-sized memory regions. Fine-grain access control forms the basis of efficient cache-coherent shared memory. This paper focuses on low-cost implementations that require little or no additional hardware. These techniques permit efficient implementation of shared memory on a wide range of parallel systems, thereby providing shared-memory codes with a portability ...

13 Multi-level shared caching techniques for scalability in VMP-M/C

D. R. Cheriton, H. A. Goosen, P. D. Boyle

April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th**

annual international symposium on Computer architecture ISCA '89, Volume 17 Issue 3
Publisher: ACM Press

Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, citings, index terms

The problem of building a scalable shared memory multiprocessor can be reduced to that of building a scalable memory hierarchy, assuming interprocessor communication is handled by the memory system. In this paper, we describe the VMP-MC design, a distributed parallel multi-computer based on the VMP multiprocessor design, that is intended to provide a set of building blocks for configuring machines from one to several thousand processors. VMP-MC uses a memory hierarchy based on shared caches ...

14 Coupling compiler-enabled and conventional memory accessing for energy efficiency
Raksit Ashok, Saurabh Chheda, Csaba Andras Moritz
May 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 2
Publisher: ACM Press
Full text available: pdf(1.41 MB) Additional Information: full citation, abstract, references, index terms

This article presents Cool-Mem, a family of memory system architectures that integrate conventional memory system mechanisms, energy-aware address translation, and compiler-enabled cache disambiguation techniques, to reduce energy consumption in general-purpose architectures. The solutions provided in this article leverage on interlayer tradeoffs between architecture, compiler, and operating system layers. Cool-Mem achieves power reduction by statically matching memory operations with energy-eff ...

Keywords: Energy efficiency, translation buffers, virtually addressed caches

15 Distributed operating systems
Andrew S. Tanenbaum, Robbert Van Renesse
December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4
Publisher: ACM Press

Full text available: pdf(5.49 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

16 Protection traps and alternatives for memory management of an object-oriented language
Antony L. Hosking, J. Eliot B. Moss
December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles SOSP '93**, Volume 27 Issue 5
Publisher: ACM Press

Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, references, citings, index terms

Many operating systems allow user programs to specify the protection level (inaccessible, read-only, read-write) of pages in their virtual memory address space, and to handle any protection violations that may occur. Such page-protection techniques have been exploited by several user-level algorithms for applications including generational garbage collection and persistent stores. Unfortunately, modern hardware has made efficient handling of

page protection faults more difficult. Moreover, page- ...

17 Design of the Mneme persistent object store

J. Eliot B. Moss
April 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 2
Publisher: ACM Press

Full text available: pdf(3.22 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The Mneme project is an investigation of techniques for integrating programming language and database features to provide better support for cooperative, information-intensive tasks such as computer-aided software engineering. The project strategy is to implement efficient, distributed, persistent programming languages. We report here on the Mneme persistent object store, a fundamental component of the project, discussing its design and initial prototype. Mneme stores objects

18 A structural view of the Cedar programming environment

Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann
August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4
Publisher: ACM Press

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

19 RISCY patents

David A. Patterson
September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4
Publisher: ACM Press
Full text available: pdf(1.83 MB) Additional Information: full citation, index terms

20 Fault Tolerant Operating Systems

Peter J. Denning
December 1976 **ACM Computing Surveys (CSUR)**, Volume 8 Issue 4
Publisher: ACM Press
Full text available: pdf(2.69 MB) Additional Information: full citation, references, citings, index terms